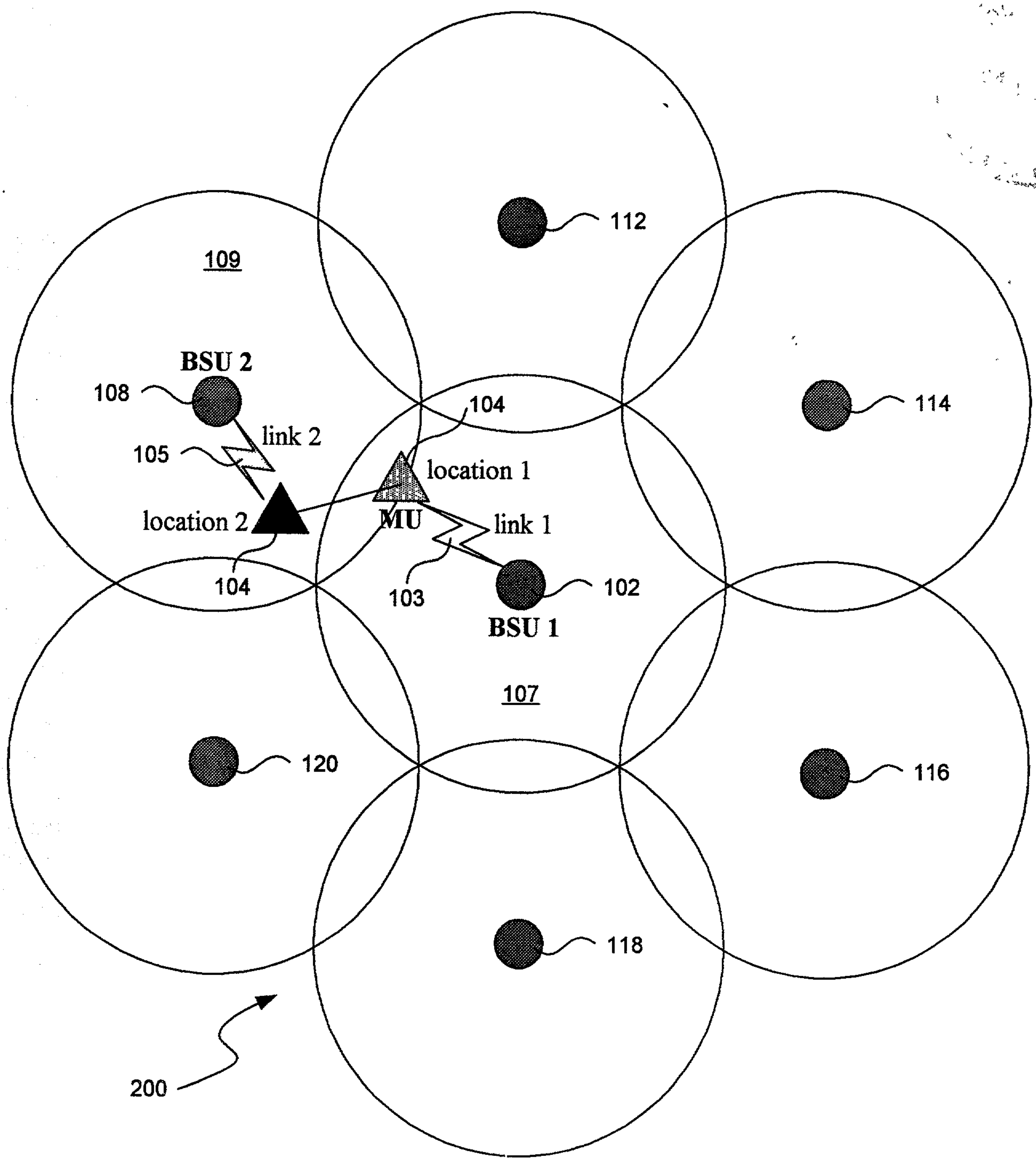

112
109
BSU 2
108
104
link 2
105
location 1
location 2
MU
link 1
104
103
102
BSU 1
114
107
120
116
118
200

Fig. 1

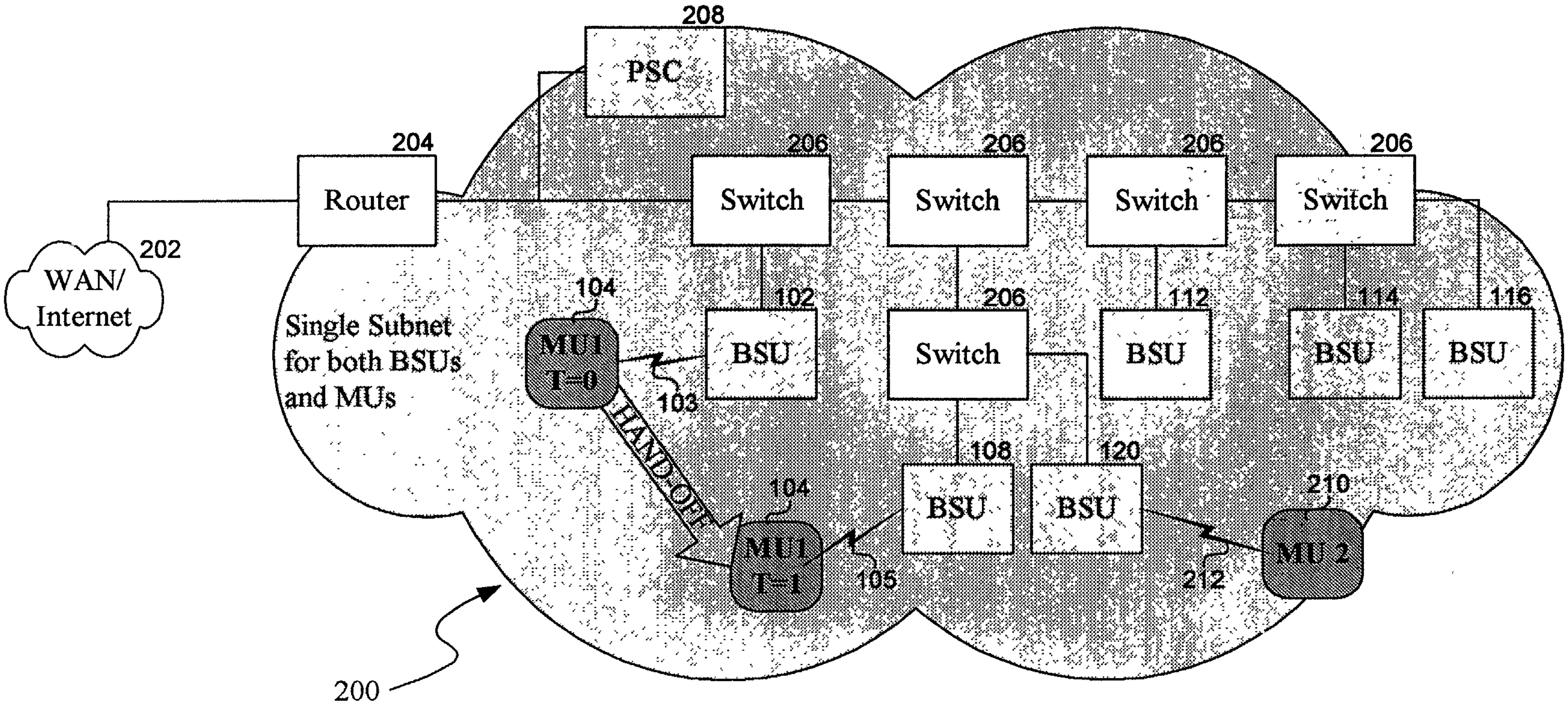

208
PSC
204
Router
206
Switch
206
Switch
206
Switch
206
Switch
202
WAN/
Internet
Single Subnet
for both BSUs
and MUs
104
MU1
T=0
103
102
BSU
206
Switch
112
BSU
114
BSU
116
BSU
HAND-OFF
104
MU1
T=1
105
108
BSU
120
BSU
212
210
MU 2
200

Fig. 2

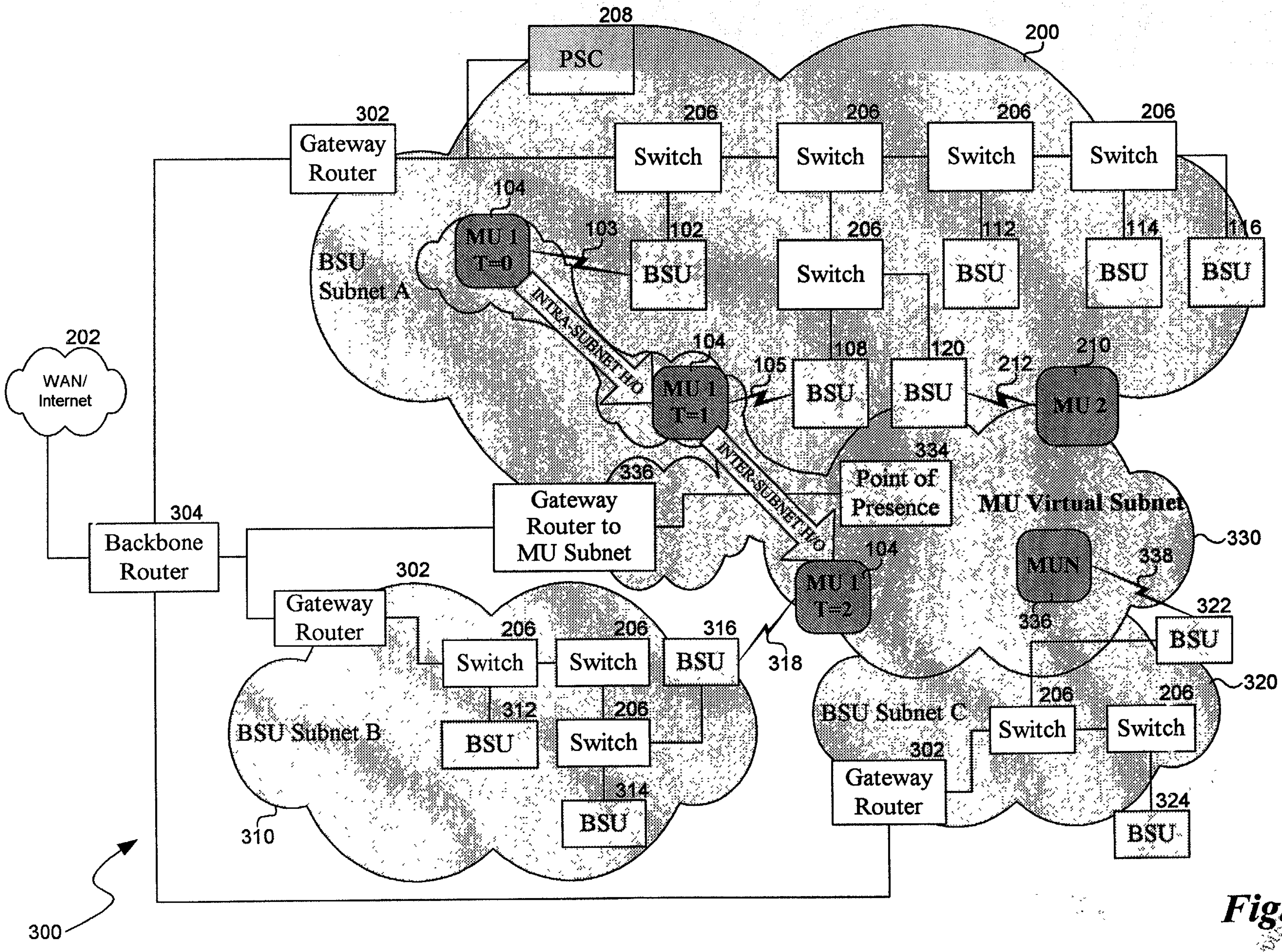

208
PSC
200
302
Gateway Router
206
Switch
206
Switch
206
Switch
206
Switch
104
MU 1 T=0
103
102
BSU
206
Switch
112
BSU
114
BSU
116
BSU
BSU Subnet A
INTRA-SUBNET H/O
104
MU 1 T=1
105
108
BSU
120
BSU
212
210
MU 2
202
WAN/ Internet
336
INTER-SUBNET H/O
334
Point of Presence
MU Virtual Subnet
304
Backbone Router
Gateway Router to MU Subnet
104
MU 1 T=2
MUN
338
330
302
Gateway Router
206
Switch
206
Switch
316
BSU
318
336
322
BSU
312
BSU
206
Switch
BSU Subnet B
314
BSU
310
BSU Subnet C
206
Switch
206
Switch
320
302
Gateway Router
324
BSU
300

Fig. 3

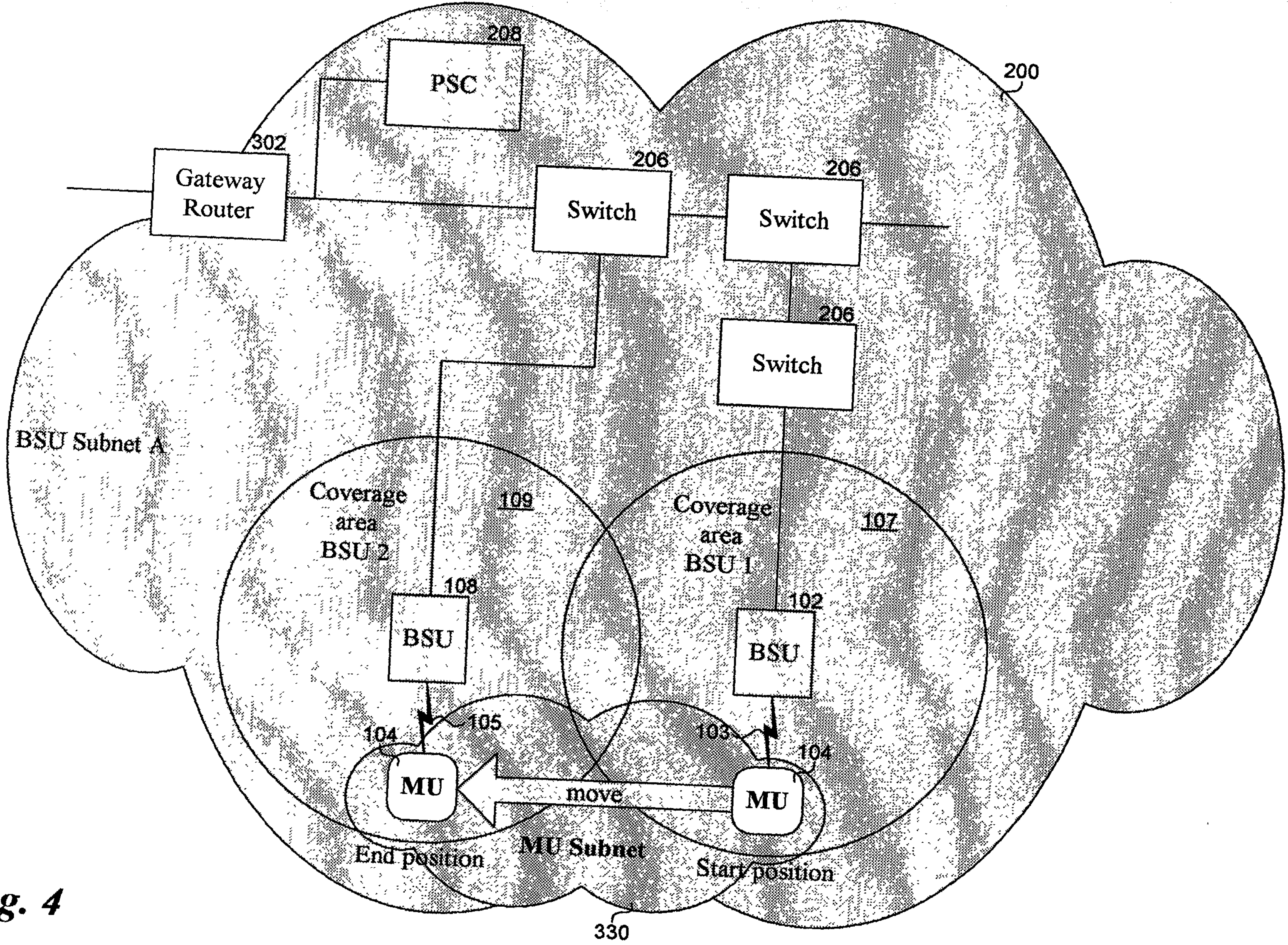

208
PSC
200
302
Gateway Router
206
Switch
206
Switch
206
Switch
BSU Subnet A
Coverage area BSU 2
109
Coverage area BSU 1
107
108
BSU
102
BSU
104
105
103
104
MU
move
MU
End position
MU Subnet
Start position
330

Fig. 4

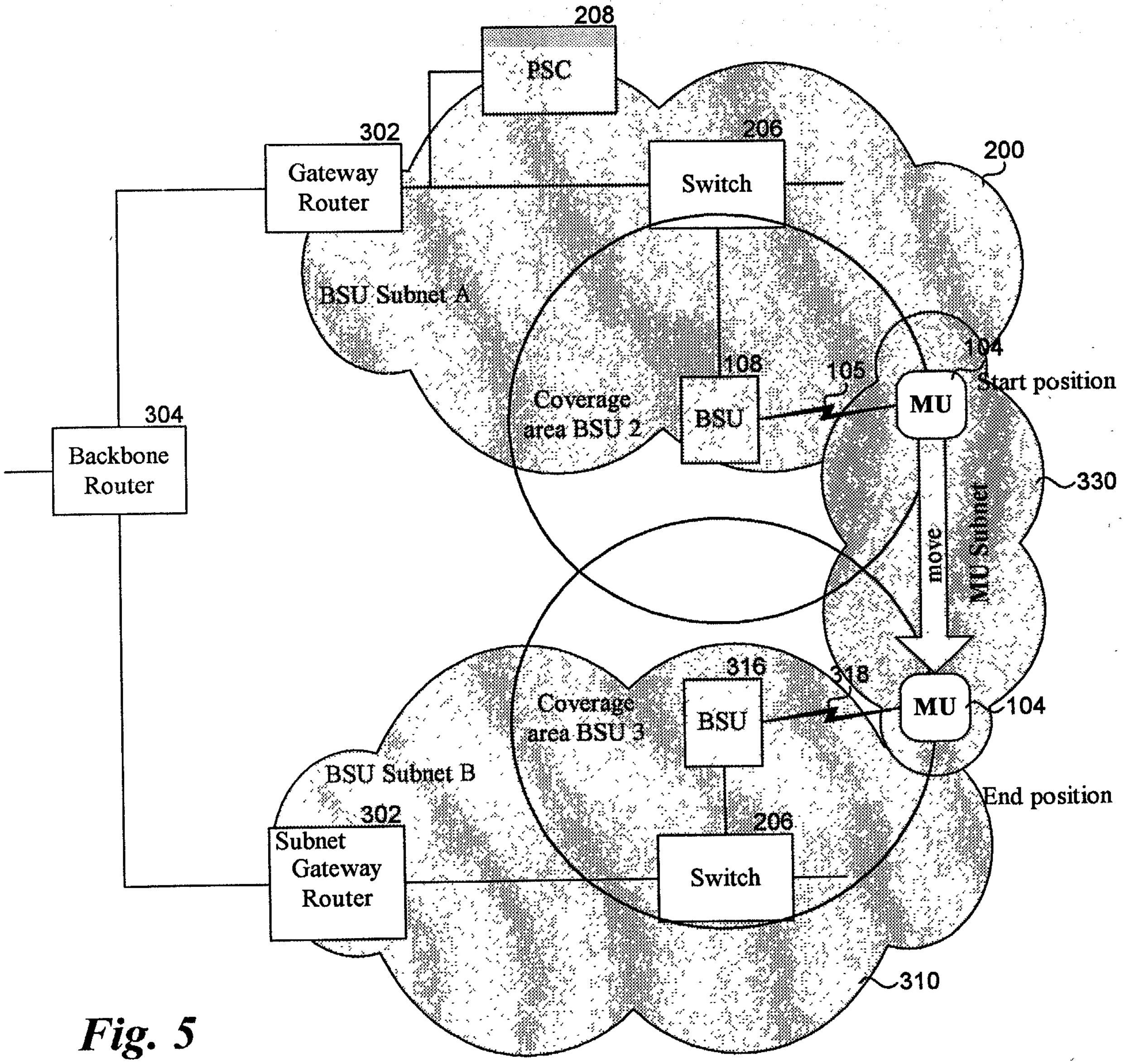

208
PSC
302
Gateway Router
206
Switch
200
BSU Subnet A
108
BSU
105
104
MU
Start position
Coverage area BSU 2
304
Backbone Router
330
move
MU Subnet
316
BSU
318
MU
104
Coverage area BSU 3
BSU Subnet B
End position
302
Subnet Gateway Router
206
Switch
310

Fig. 5

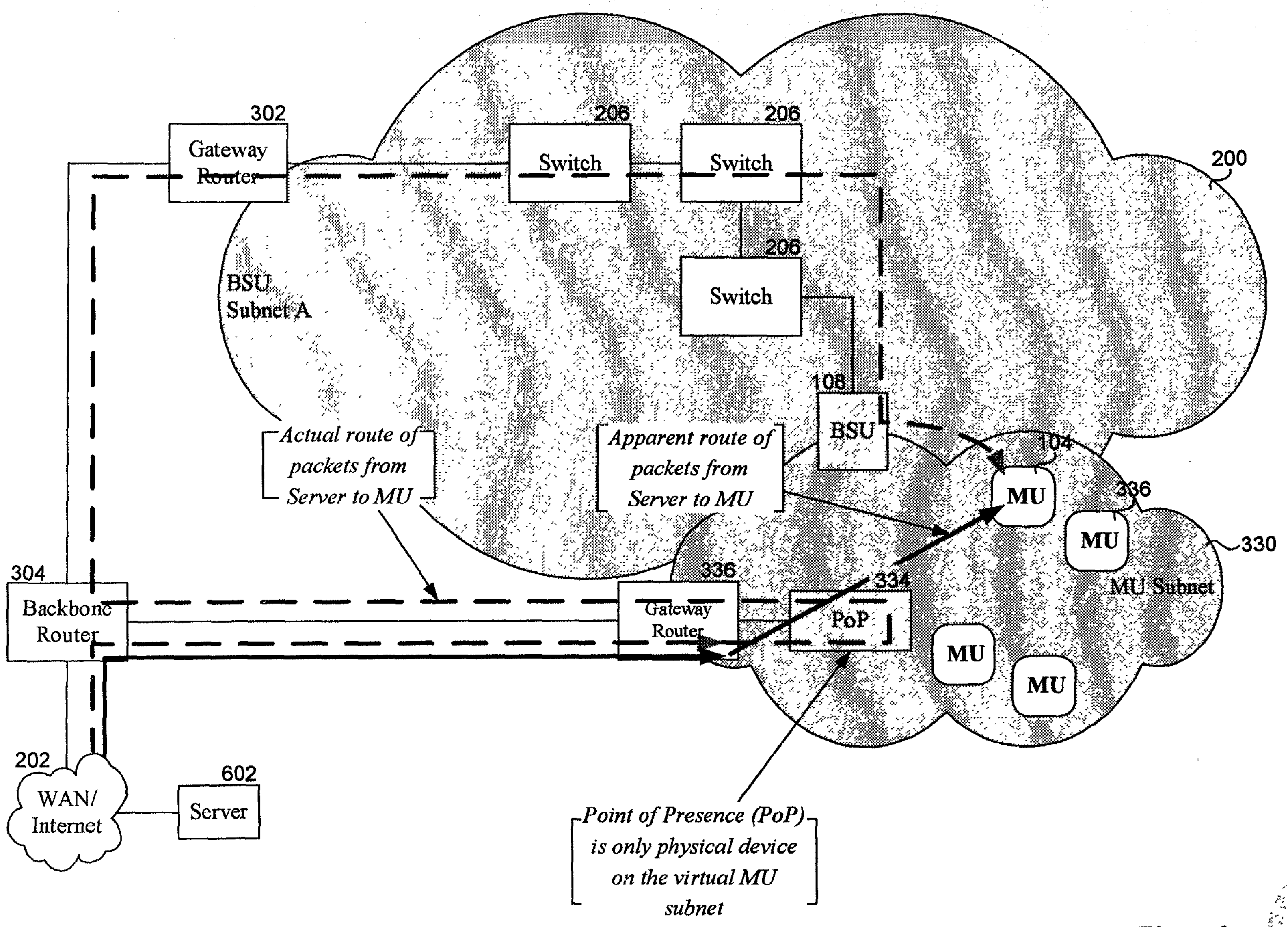

302
Gateway Router
206
Switch
206
Switch
206
Switch
200
BSU Subnet A
108
BSU
104
MU
336
MU
330
MU Subnet
MU
MU
Actual route of packets from Server to MU
Apparent route of packets from Server to MU
304
Backbone Router
336
Gateway Router
334
PoP
202
WAN/ Internet
602
Server
Point of Presence (PoP) is only physical device on the virtual MU subnet

Fig. 6

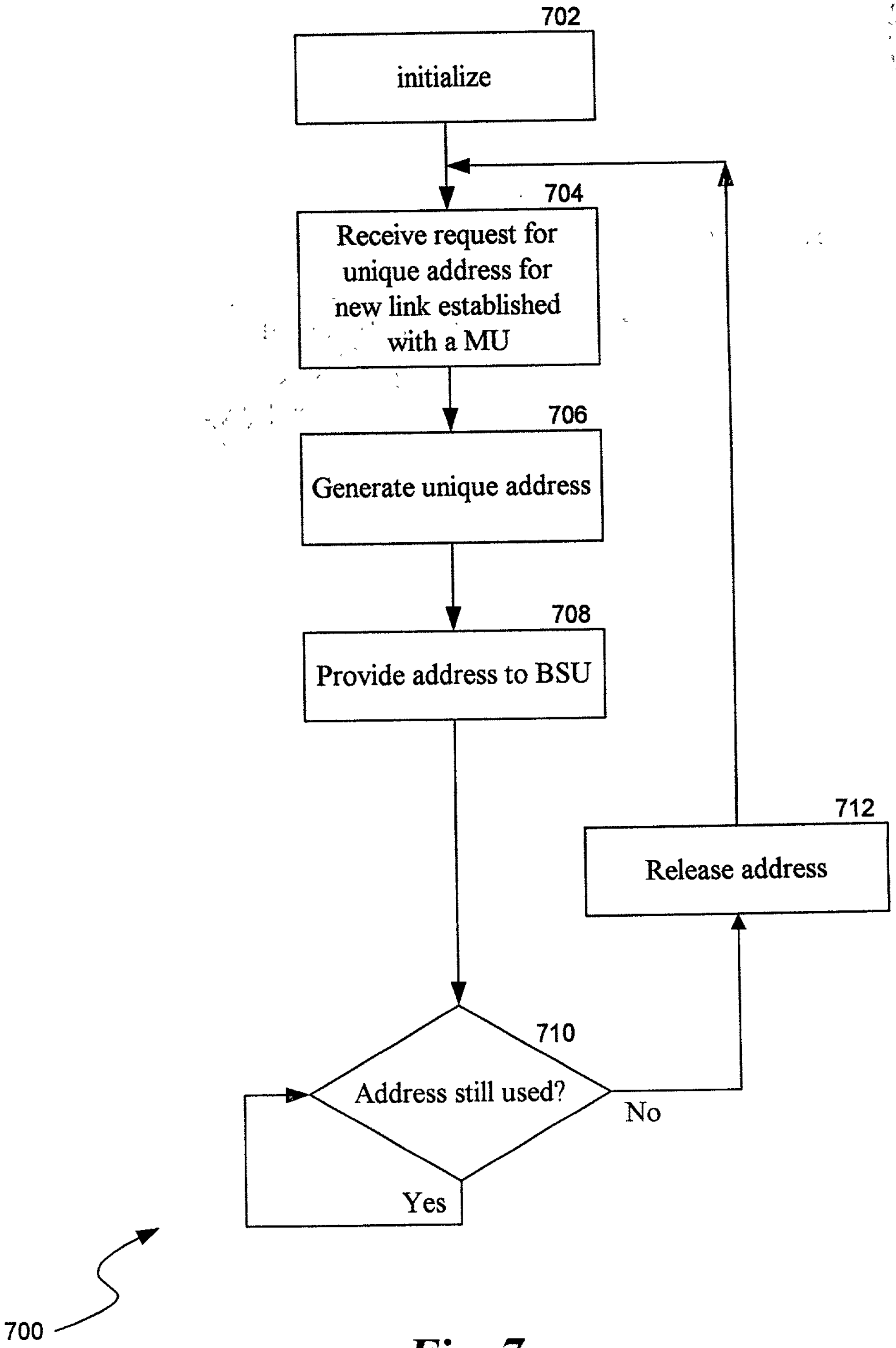

702
initialize
704
Receive request for unique address for new link established with a MU
706
Generate unique address
708
Provide address to BSU
710
Address still used?
Yes
No
712
Release address
700

Fig. 7

Mobility Management (highlighted). LMS is Link Management Support

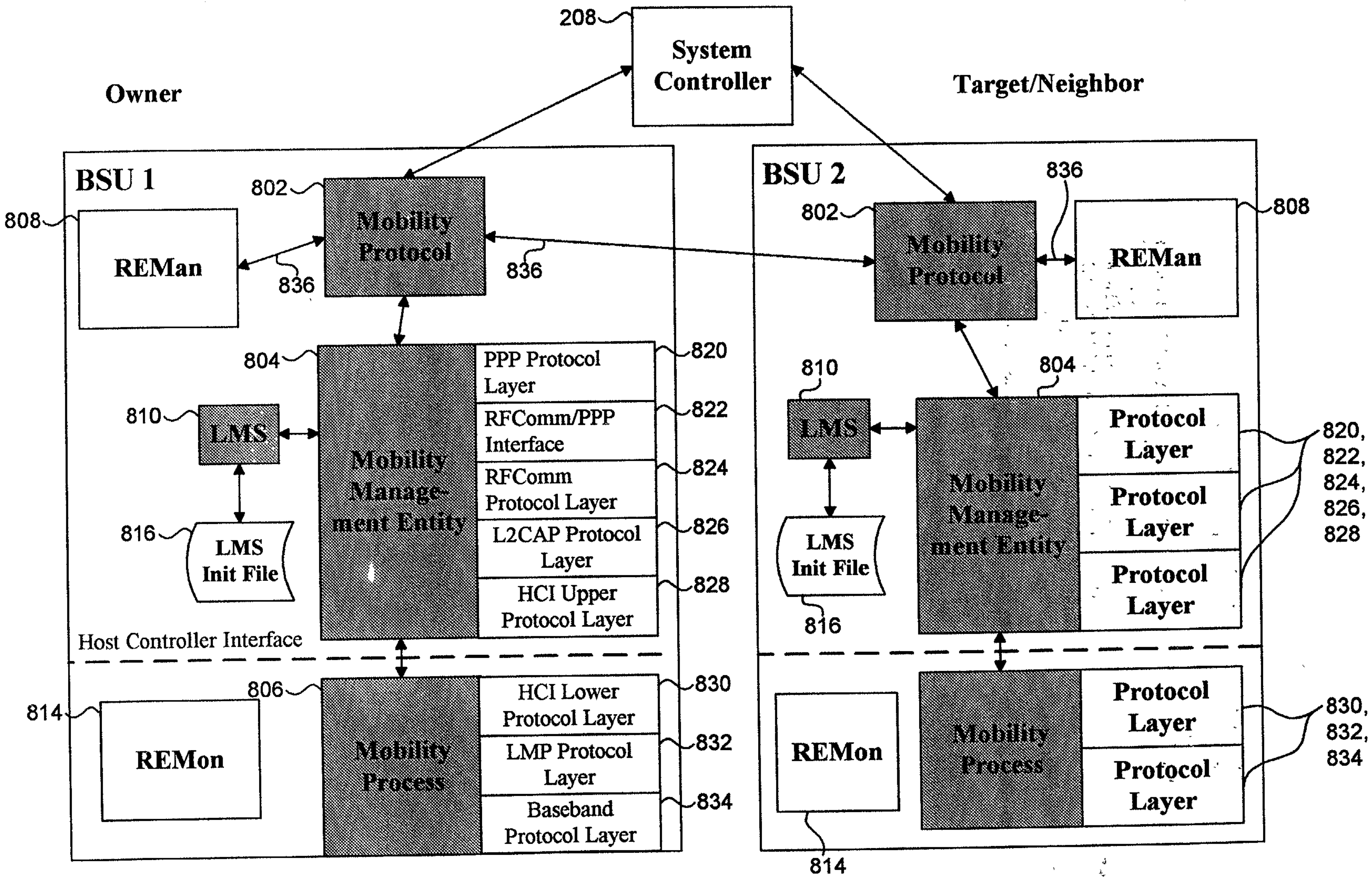


Fig. 8

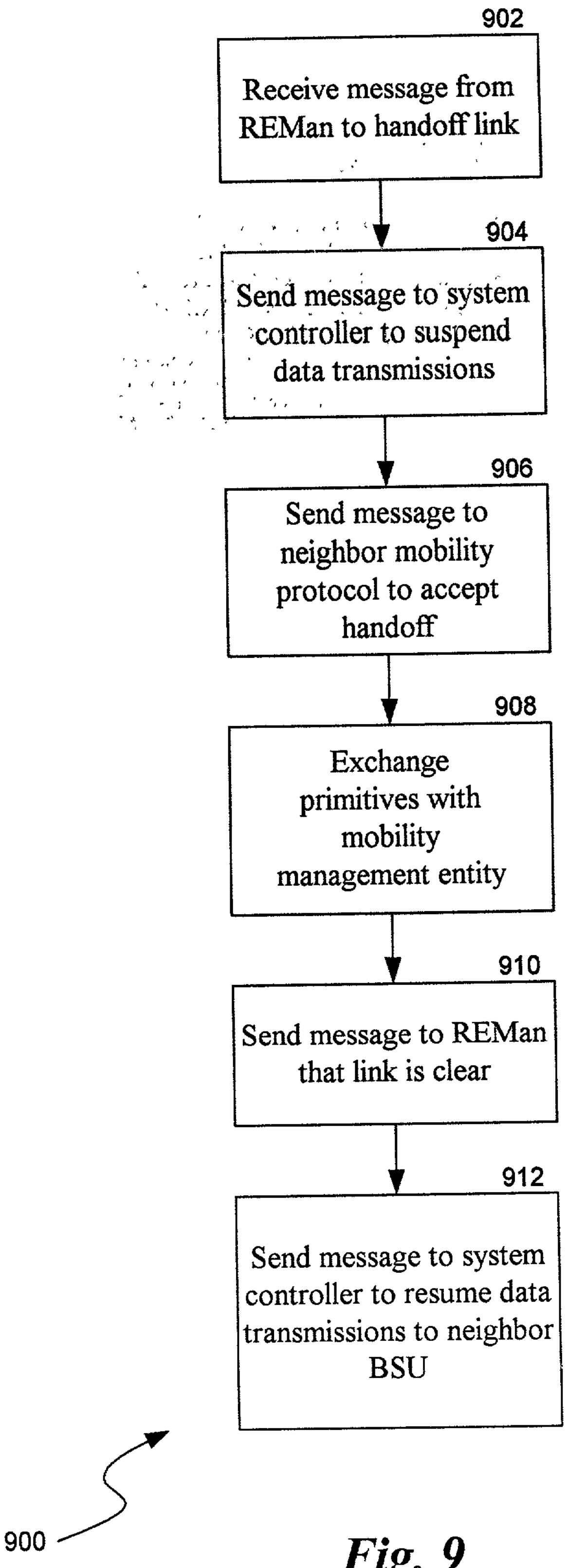

902
Receive message from REMan to handoff link
904
Send message to system controller to suspend data transmissions
906
Send message to neighbor mobility protocol to accept handoff
908
Exchange primitives with mobility management entity
910
Send message to REMan that link is clear
912
Send message to system controller to resume data transmissions to neighbor BSU
900

Fig. 9

Link Context Record

Virtual Bluetooth device address ("BD_Addr")	<unique 48 bit address>	1002
BSU system clock offset (CLK_OFFSET)	<offset value>	1004
Active Member Address for MU	<integer 0:7>	1006
Encryption keys (optional)	<integer>	1008
BSU's IP Address	<local IP Address>	1010
Mode and timing parameters	<Mode: Sniff, Hold, Park>, <time>	1012
Mobile Unit ID ("BD_Addr")	<unique 48 bit address>	1014
Channel ID	<16 bit value>	1016
Link Initialization Time	<date, time>	1018

1000

Fig. 10

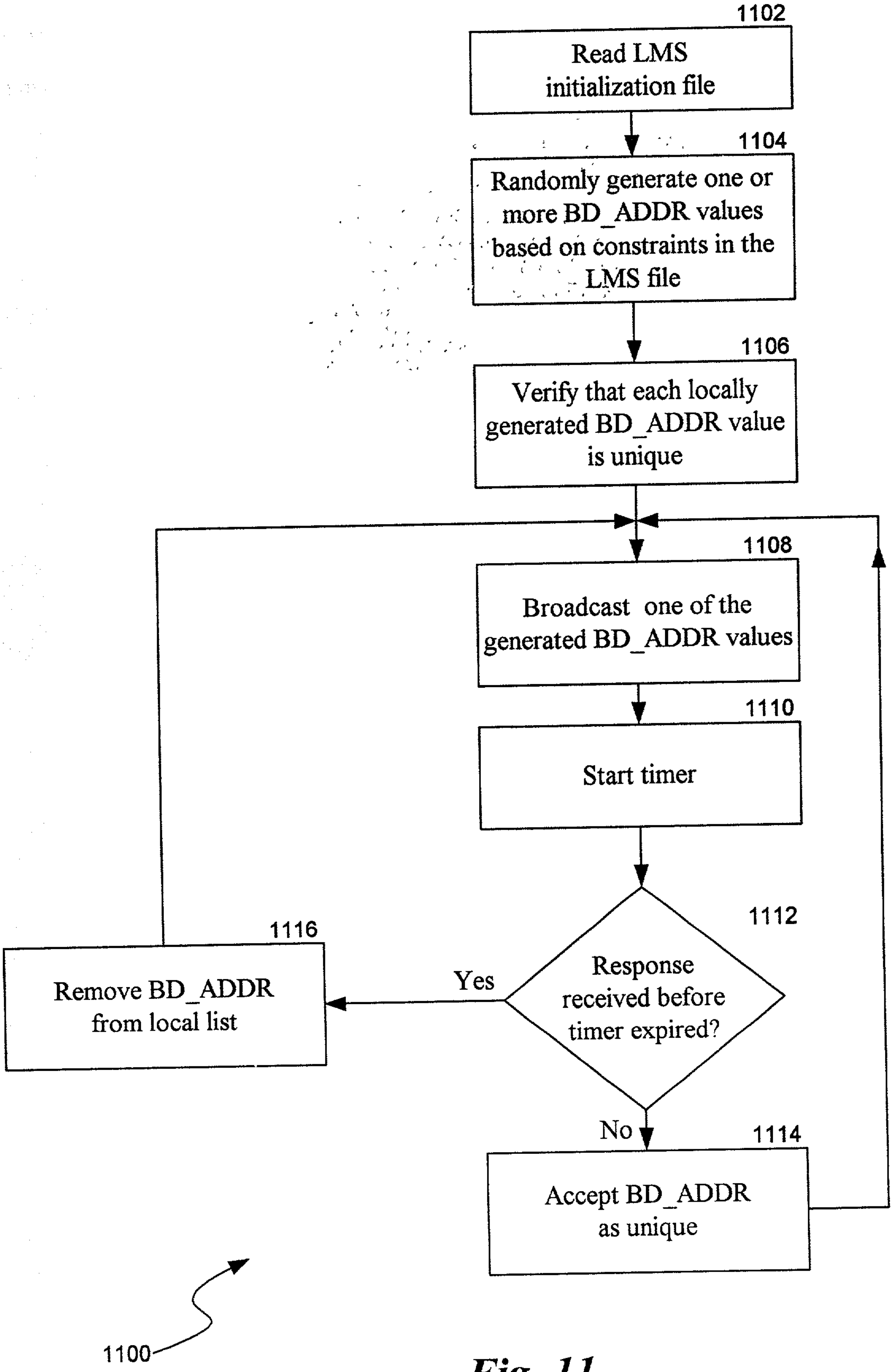
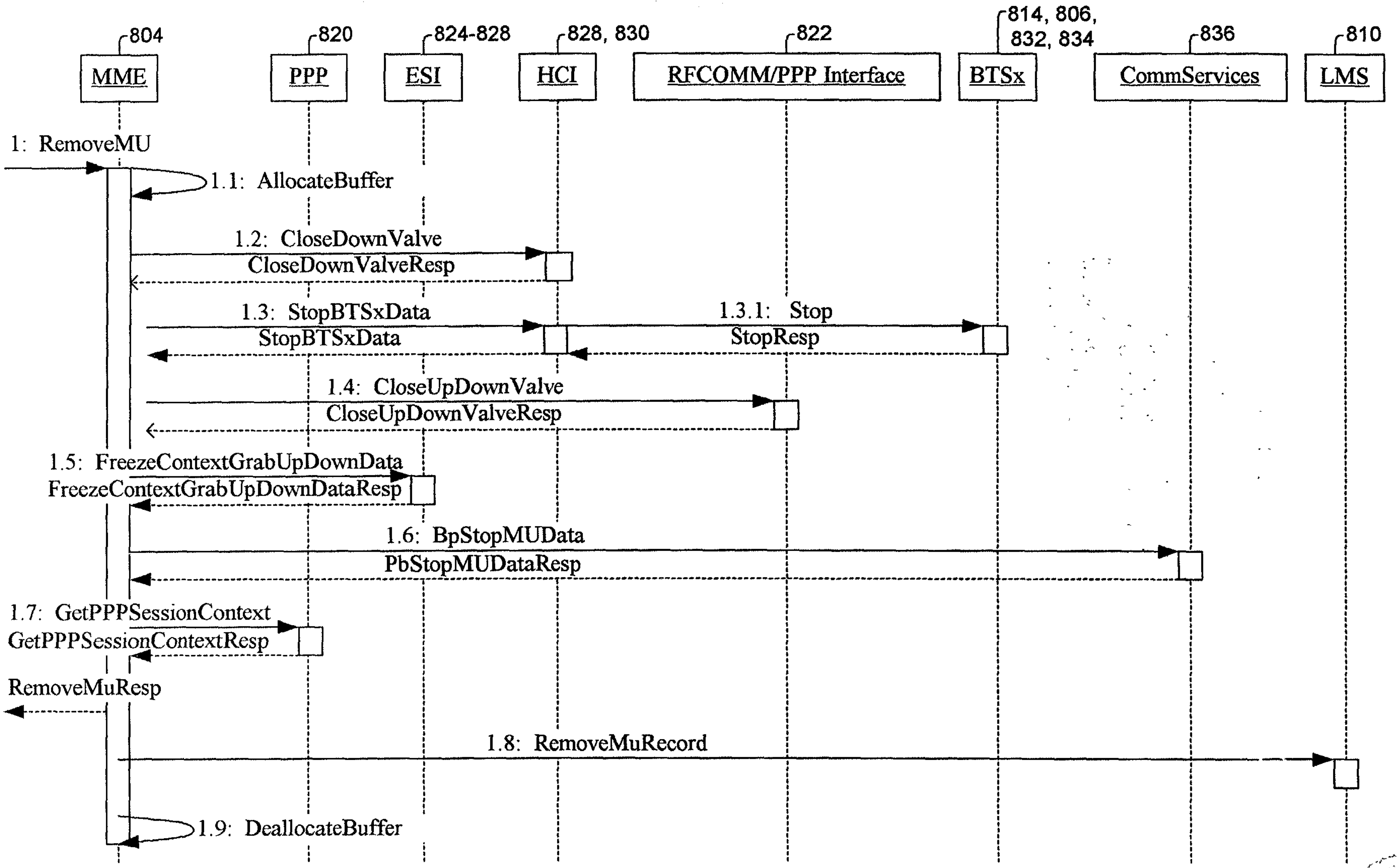

1102
Read LMS initialization file
1104
Randomly generate one or more BD_ADDR values based on constraints in the LMS file
1106
Verify that each locally generated BD_ADDR value is unique
1108
Broadcast one of the generated BD_ADDR values
1110
Start timer
1112
Response received before timer expired?
Yes
1116
Remove BD_ADDR from local list
No
1114
Accept BD_ADDR as unique
1100

Fig. 11

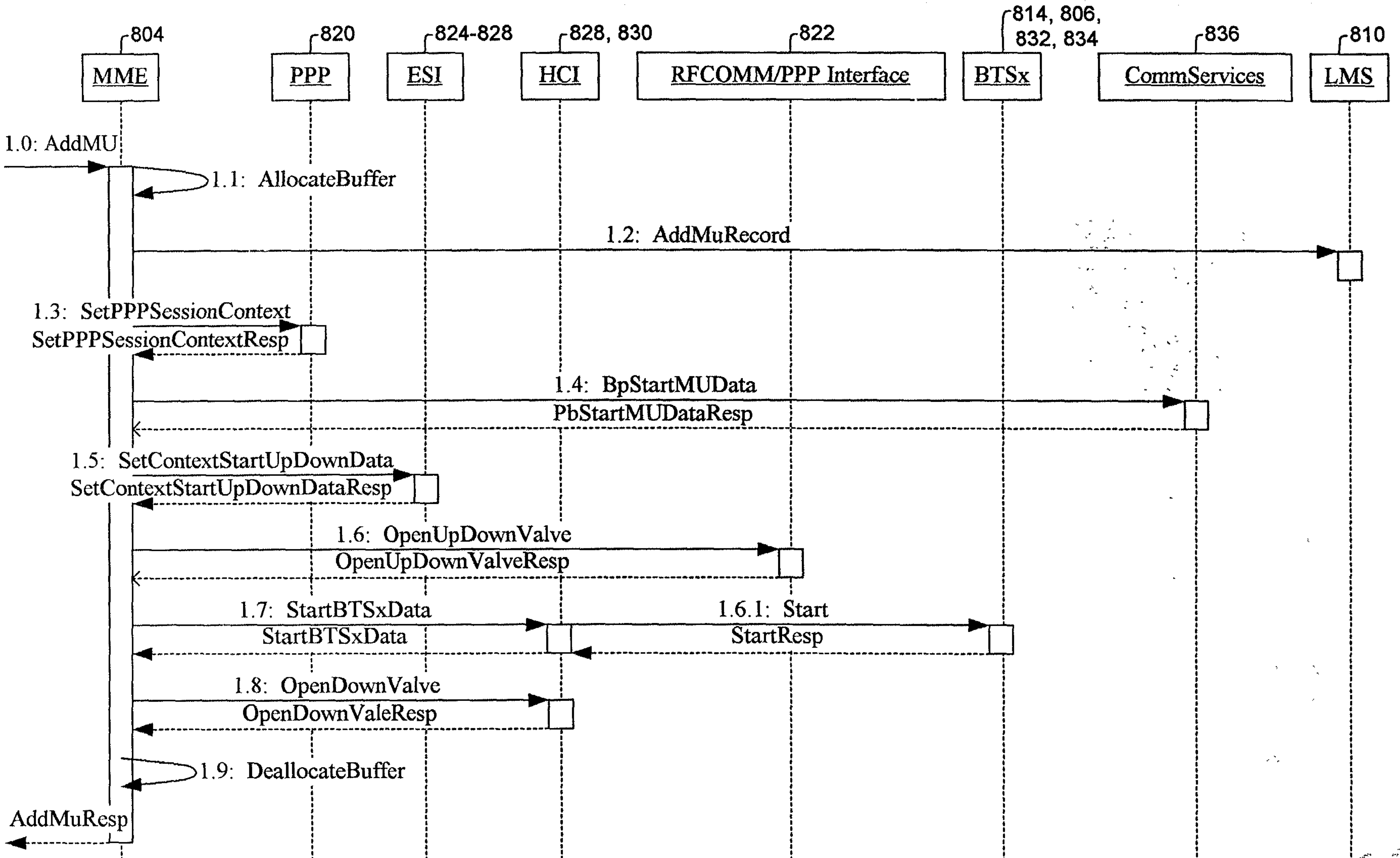
MME Functionality: Owner BSU Performing a Handoff



The MME on the owner BSU receives a RemoveMU call. That calls starts a chain of messages that will result in the buffered data flowing to the BTSx getting captured and sent to the target BSU. Along with that data will come the session context.

Fig. 12

MME Functionality: Target BSU Receiving a Handoff



The MME gets its AddMu method called. This means that a MU session must be established. This includes any data that is being sent to the MU and the session context.

Fig. 13

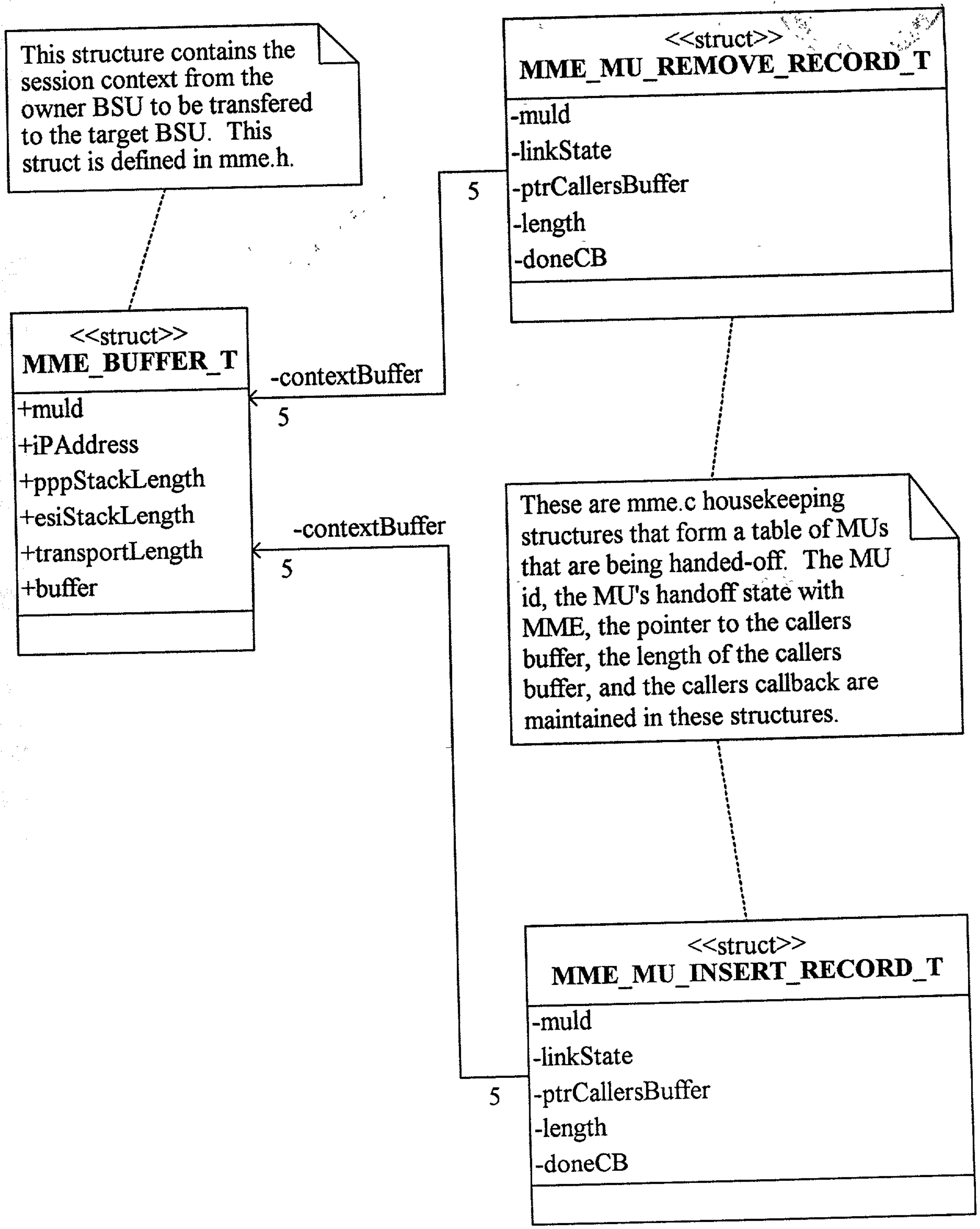

This structure contains the session context from the owner BSU to be transfered to the target BSU. This struct is defined in mme.h.
<<struct>>
MME_MU_REMOVE_RECORD_T
-muId
-linkState
-ptrCallersBuffer
-length
-doneCB
5
<<struct>>
MME_BUFFER_T
+muId
+iPAddress
+pppStackLength
+esiStackLength
+transportLength
+buffer
-contextBuffer
5
-contextBuffer
5
These are mme.c housekeeping structures that form a table of MUs that are being handed-off. The MU id, the MU's handoff state with MME, the pointer to the callers buffer, the length of the callers buffer, and the callers callback are maintained in these structures.
<<struct>>
MME_MU_INSERT_RECORD_T
-muId
-linkState
-ptrCallersBuffer
-length
-doneCB
5

Fig. 14